

## BAB II

### LANDASAN TEORI

#### 2.1 Data Mining

**Data Mining** adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa informasi yang selama ini tidak diketahui secara manual dari suatu basis data dengan melakukan penggalian pola-pola dari data, dengan tujuan untuk memanipulasi data menjadi informasi yang lebih berharga yang diperoleh dengan cara mengekstraksi dan mengenali pola yang penting atau menarik dari data yang terdapat dalam basisdata [2].

**Data Mining** juga memiliki istilah lain yaitu *knowledge-discovery in database (KDD)*. Data Mining atau KDD bertujuan untuk memanfaatkan data yang terdapat dalam basis data dengan mengolah data tersebut sehingga di dapatkan suatu informasi yang berguna. Data Mining sendiri memiliki empat akar ilmu yaitu sebagai berikut :

1. Statistik

Bidang ini merupakan akar paling tua, tanpa ada statistik maka data mining mungkin tidak ada. Dengan menggunakan statistik klasik ternyata data yang diolah dapat diringkas dalam apa yang umum dikenal sebagai *exploratory data analysis* (EDA). EDA berguna untuk mengidentifikasi hubungan sistematis antar variable/fitur ketika tidak ada cukup informasi alami yang dibawanya.

2. Kecerdasan buatan atau *Artificial Intelligent* (AI)

Teorinya dibangun berdasarkan teknik heuristik sehingga AI berkontribusi terhadap teknik pengolahan informasi berdasarkan pada model penalaran manusia. Salah satu cabang dari AI adalah pembelajaran mesin atau *machine learning*, merupakan disiplin ilmu yang paling penting yang direperesentasikan dalam pembangunan data mining, menggunakan teknik dimana sistem komputer belajar dengan ‘pelatihan’.

3. Pengenalan pola atau *Pattern Recognition*

Data mining juga menjadi turunan bidang pengenalan pola, tetapi hanya mengolah data dari basis data.

4. Sistem basis data

Akar bidang ilmu keempat dari data mining yang menyediakan informasi berupa data yang akan di gali menggunakan metode-metode yang terdapat di data mining [3].

Ada beberapa metode dalam data mining, namun yang paling sering di gunakan adalah klasifikasi. Klasifikasi mempelajari pola-pola dari data historis yaitu berupa sekumpulan informasi seperti ciri-ciri , variable-variabel, fitur-fitur serta berbagai karaktetistik dengan tujuan untuk menempatkan object-object baru ke dalam kelompok atau kelas masing-masing. Metode lain dalam data mining adalah *clustering*, *association*, *regression*, *Forecasting*, *sequence analysis*, dan *deviation analysis*. Selain memiliki beberapa metode, data mining juga memiliki beberapa jenis algoritma sebagai berikut :

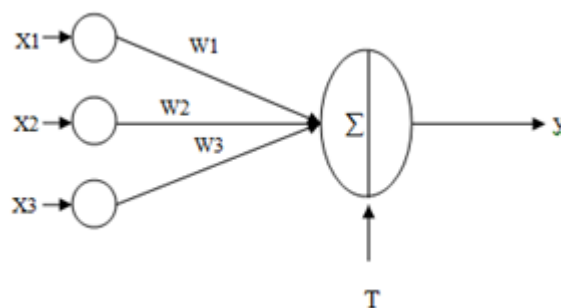
1. Metode *Naïve Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik (memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya).
2. *K-nearest neighbor* (K-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut.
3. *Decision Tree* merupakan sebuah metode pengklasifikasian dengan langkah-langkah menentukan atribut yang paling tinggi Information Gainnya untuk ditetapkan sebagai level tertinggi yang mempengaruhi data. Untuk Information Gainnya akan didapatkan dari pengurangan *Entropy*.
4. *Artificial Neural Network* (ANN) adalah suatu sistem pemrosesan informasi yang mencoba meniru kinerja otak manusia [4].

## 2.2 Konsep Artificial Neural Network (ANN)

Ketika terlahir, manusia memiliki otak yang terstruktur yang sangat menakjubkan karena kemampuannya dalam membentuk sendiri aturan-aturan/pola berdasarkan pengalaman yang telah diterima sebelumnya. *Artificial neural network*(ANN) atau jaringan syaraf tiruan sendiri adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi pada otak manusia. Tujuan dari dibuatnya *Artificial neural network* adalah membuat sistem yang dapat belajar sendiri sesuai dengan data dan kondisi yang telah di berikan [5]. Dengan menganalogikan ANN yang meniru cara kerja otak pada manusia, maka dapat di asumsikan sebagai berikut:

- Pemrosesan informasi terjadi pada banyak elemen sederhana yang disebut *neuron* (dalam otak manusia dikenal dengan *axon*) yang berisi *adder* dan *fungsi aktivasi*.
- Sinyal dikirimkan antara neuron-neuron melalui penghubung-penghubung(dalam otak manusia dikenal dengan *sinapsis*).
- Penghalang antar neuron memiliki bobot yang akan memperkuat dan memperlemah sinyal (dalam otak manusia dikenal dengan *dendrit*).
- Untuk menentukan output, setiap neuron menggunakan *fungsi aktivasi* (biasanya bukan fungsi linier) yang dikenakan pada jumlah input yang diterima. Besarnya output selanjutnya dibandingkan dengan suatu ambang batas (*threshold*)[6].

Gambar desain ANN secara umum ditunjukkan oleh gambar 2.1

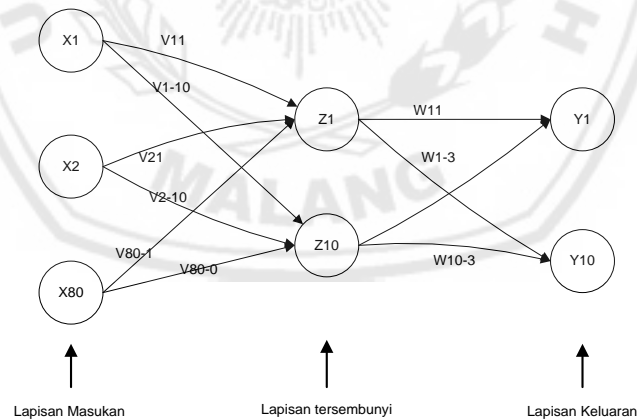


Pada gambar tersebut, dijelaskan sebagai berikut :

1. Vektor masukan terdiri dari sejumlah nilai (fitur), vector masukan tersebut terdiri dari 3 nilai ( $x_1, x_2, x_3$ ) sebagai fitur dalam vektor yang akan diproses dalam ANN.
2. Masing-masing nilai masukan melewati sebuah hubungan berbobot  $w(w_1, w_2, w_3)$ , kemudian semua nilai digabungkan.
3. Nilai gabungan tersebut kemudian diproses oleh fungsi aktivasi untuk menghasilkan sinyal  $y$  sebagai keluaran.
4. Fungsi aktivasi menggunakan sebuah nilai ambang batas atau *threshold* ( $T$ ) untuk membatasi nilai keluaran agar selalu dalam batas nilai yang ditetapkan [7].

### 2.2.1 Arsitektur Artificial Neural Network (ANN)

Arsitektur Artificial neural network (ANN) memiliki beberapa lapisan yaitu lapisan input (input layer), lapisan tersembunyi (hidden layer), serta output layer atau sering di sebut juga error backpropagation (multilayer perceptron) [8]. Pada gambar 2.3 dapat di lihat arsitektur Artificial neural network (ANN).



Gambar 2.2 Arsitektur Umum Artificial Neural Network (ANN)

Keterangan:

1.  $X$  = Masukan (*Input*)
2.  $V$  = Bobot pada lapisan tersembunyi
3.  $Z$  = Lapisan tersembunyi

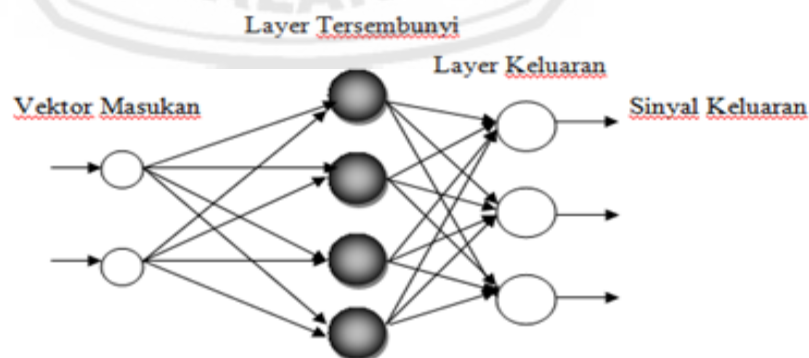
4.  $W$  = Bobot pada lapisan keluaran
5.  $Y$  = Keluaran (*Output*)

### 2.3 Algoritma Backpropagation

*ANN Perceptron* merupakan salah satu jenis ANN dengan layer tunggal. *Artificial Neural Network* (ANN) dengan layer tunggal memiliki keterbatasan dalam pengenalan pola. Penemuan *backpropagation* yang terdiri dari beberapa layer membuat ANN kembali membuka cakrawala untuk terus dikembangkan. Backpropagation melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

Cara kerja metode *backpropagation* pada umumnya adalah sebagai berikut : tahap awal adalah memberikan pola masukan sebagai pola input kepada *Artificial Neural Network* (ANN) yang akan di proses dan selanjutnya menuju ke unit-unit keluaran. Kemudian unit-unit keluaran tersebut akan memberikan tanggapan yang disebut sebagai keluaran *Artificial Neural Network*[9].

Gambar arsitektur jaringan *Artificial Neural Network* (ANN) *backpropagation* dibawah ini:



Gambar 2.3 Arsitektur Jaringan *Artificial Neural Network* (ANN) *backpropagation*

Contoh arsitektur jaringan *Artificial Neural Network* (ANN) *backpropagation* pada gambar 2.3. Pada gambar tersebut terdapat satu hidden layer yang memiliki empat neuron, dan satu layer keluaran dengan tiga neuron. Terdapat satu layer lagi dalam ANN, yaitu layer masukan berupa vector masukan. Karena tidak adanya proses komputasi pada layer ini, maka yang dilakukan hanya meneruskan sinyal/vektor masukan yang telah diterima ke layer didepannya.

Setiap layer dalam MLP (*multilayer perceptron*) masing-masing mempunyai fungsi khusus. Layer masukan berfungsi menerima sinyal/vektor masukan dari luar dan mendistribusikannya ke semua neuron dalam hidden layer. Layer keluaran menerima sinyal keluaran (atau dengan kata lain stimulus pola) dari hidden layer dan memunculkan sinyal/nilai/kelas keluaran dari keseluruhan jaringan.

Neuron dalam hidden layer mendeteksi fitur-fitur yang tersembunyi, sedangkan bobot dari neuron dalam hidden layer merepresentasikan fitur tersembunyi dalam vector masukan. Fitur-fitur tersembunyi ini kemudian digunakan oleh layer keluaran dalam menentukan pola/kelas keluaran. Dengan satu hidden layer dapat merepresentasikan sembarang fungsi kontinu dari sinyal masukan. Sementara dengan dua hidden layer, fungsi diskontinu pun dapat direpresentasikan.

Hidden layer menyembunyikan keluaran yang diinginkan. Neuron dalam hidden layer tidak dapat diamati melalui perilaku masukan/keluaran jaringan secara keseluruhan. Selain itu, juga tidak ada cara yang jelas untuk mengetahui apa keluaran yang diinginkan oleh hidden layer itu sendiri.

Berikut adalah algoritma pelatihan *backpropagation* adalah sebagai berikut:

### **Langkah 1: inisialisasi**

Inisialisasi semua bobot pada hidden layer dan layer keluaran, lalu tetapkan fungsi aktivasi yang digunakan untuk setiap layer. Untuk

inisialisasi semua bobot bias menggunakan bilangan acak dalam jangkauan  $[-0.5, 0.5]$  atau menggunakan distribusi seragam dalam jangkauan kecil (Haykin, 1999):

$$\left(-\frac{2.4}{Fi}, +\frac{2.4}{Fi}\right) \dots\dots\dots(a)$$

$f_i$  adalah jumlah neuron masukan neuron I dalam ANN.

### Langkah 2: Aktivasi

Mengaktivasi jaringan dengan menerapkan masukan  $x_1(p)$ ,  $x_2(p)$ , ...,  $x_n(p)$ , dan keluaran yang diharapkan  $y_{d1}(p)$ ,  $y_{d2}(p)$ , ...,  $y_{dn}(p)$ .

a. Hitung keluaran yang didapatkan dari neuron dalam hidden layer:

$$v_j(p) = \sum_{i=1}^r x_i(p) \cdot w_{ij}(p) \dots\dots\dots(b)$$

$$y_j(p) = \frac{1}{1 + e^{-v_k(p)}} \dots\dots\dots(c)$$

$r$  adalah jumlah masukan (fitur) pada neuron  $j$  dalam hidden layer.

b. Hitung keluaran yang didapatkan dari neuron dalam layer keluaran.

$$v_k(p) = \sum_{j=1}^m x_j(p) \cdot w_{jk}(p) \dots\dots\dots(d)$$

$$y_k(p) = \frac{1}{1 + e^{-v_k(p)}} \dots\dots\dots(e)$$

$m$  adalah jumlah masukan pada neuron  $k$  dalam layer keluaran

### Langkah 3: Perbarui bobot

Bobot diperbarui pada saat error ditambahkan balik dalam ANN, error yang dikembalikan sesuai dengan arah keluarnya sinyal keluaran.

a. Hitung gradien error untuk neuron dalam layer keluaran:

$$e_k(p) = y_{dk}(p) - y_k(p) \dots\dots\dots (f)$$

$$\hat{o}_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p) \dots\dots\dots (g)$$

Hitung koreksi bobot :

$$\Delta w_{jk}(p) = \eta \times y_j(p) \times \hat{o}_k(p) \dots\dots\dots (h)$$

Perbarui bobot pada neuron layer keluaran:

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p) \dots\dots\dots (i)$$

b. Hitung gradien error untuk neuron dalam hidden layer:

$$\hat{o}_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^l \partial_k(p) \cdot w_{jk}(p) \dots\dots\dots (j)$$

Hitung koreksi bobot:

$$\Delta w_{ij}(p) = \eta \times x_i(p) \times \hat{o}_j(p) \dots\dots\dots (k)$$

Perbarui bobot pada neuron hidden layer:

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p) \dots\dots\dots (l)$$

#### Langkah 4: iterasi

Naikkan satu untuk iterasi p, kembali kelangkah 2 dan ulangi proses tersebut sampai kriteria error tercapai [10].

## 2.4 Bahasa Pemrograman Java

Untuk menyelesaikan tugas akhir ini, penulis akan memfokuskan bahasa pemrograman yang akan di gunakan adalah bahasa java. Bill Joy mendeskripsikan bahasa java sebagai bahasa pemrograman orientasi objek yang berukuran kecil, sederhana, aman, diinterpretasi atau di optimasi secara dinamis, ber-bytecode,



netral arsitektur, mempunyai *garbage-collector*, *multithreading*, mempunyai mekanisme penanganan kecuali (*exception handling*), berbasis tipe untuk penulisan program mudah di perluas secara dinamis serta telah diperuntukkan sistem tersebar. Seperti yang di ketahui, bahasa pemrograman java saat ini yang paling banyak di gunakan oleh para *programmer* dan *software developer* untuk pengembangan berbagai macam aplikasi. Seperti aplikasi *Applet*, *desktop*, *console*, sampai aplikasi yang berskala enterprise. Berikut kelebihan dan kekurangan bahasa pemrograman java

Kelebihan bahasa pemrograman java sebagai berikut :

- **Bahasa Sederhana.** Java di rancang agar mudah dipelajari dan digunakan secara efektif. Java tidak menyediakan fitur-fitur rumit, serta banyak pekerjaan pemrogram yang mulanya harus dilakukan *manual* dikerjakan Java secara otomatis seperti dealokasi memori.
- **Bahasa Orientasi Objek.** Meskipun Java dipengaruhi bahasa-bahasa terdahulu, namun bahasa Java tidak di rancang untuk kompatibel dengan bahasa-bahasa sebelumnya. Bahasa Java sepenuhnya baru. Kebijakan ini memberi kebebasan perancang bahasa Java.
- **Bahasa *Statically Typed*.** Kebaikan lain fitur ini adalah kode lebih dapat dioptimasi untuk menghasilkan program berkinerja tinggi.
- **Bahasa Dikompilasi.** Beberapa teknologi dikembangkan agar *bytecode* berjalan cepat mendekati bahasa mesin, antara lain *Just-in-time compilation* dan *hot-spot technology*.
- **Bahasa yang Aman.** Java merupakan bahasa yang rancangannya telah memperhitungkan keamanan. Saat ini, Java telah menerapkan keamanann secara ketat namun *fleksibel*. Keamanan berdasar *file* kebijakan yang dapat diatur dan dikendalikan untuk memperoleh kemampuan maksimal program mandiri dan applet yang dipercaya.
- **Bahasa Independen terhadap Platform.** *Platform independence* adalah kemampuan program yang bekerja di sistem operasi atau sistem komputer berbeda. Bahasa Java merupakan bahasa yang secara sempurna tidak bergantung *platform*.

- **Bahasa *Multithreading*.** *Thread* adalah untuk menyatakan program komputer melakukan lebih dari satu tugas di satu waktu yang sama. Java menyediakan kakak untuk menulis program *multithread*, program mempunyai lebih dari satu *thread* eksekusi pada saat yang sama sehingga memungkinkan program menangani beberapa tugas secara konkuren. Semua aplikasi pasti sedikitnya mempunyai satu *thread* yang merepresentasikan jalur utama eksekusi.
- **Bahasa yang Didukung *Garbage Collector*.** Program Java melakukan *garbage collection* sehingga pemrogram tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pemrogram dan mengurangi atau mengeliminasi sumber kesalahan terbesar bahasa yang memungkinkan alokasi dinamis.
- **Bahasa yang Tangguh.** *Java interpreter* memeriksa semua akses sistem yang dilakukan. Program Java tidak dapat menyebabkan *crash* terhadap sistem. Java mempunyai mekanisme *exception-handling* yang ampuh. *Exception-handling* menyediakan cara pemisahan bagian penanganan kesalahan dengan bagian kode normal sehingga menuntun ke struktur kode program yang lebih bersih. Ketika kesalahan serius ditemukan, program menciptakan *exception*. *Exception* dapat ditangkap dan dikelola tanpa beresiko membuat sistem menjadi *down*.
- **Bahasa yang Mampu Diperluas.** Program Java mendukung *native method* yaitu fungsi ditulis di bahasa lain., biasanya C/C++. Dukungan *native method* memungkinkan pemrogram menulis fungsi agar dapat dieksekusi lebih cepat dibanding fungsi ekuivalen di Java. *Native method* secara dinamis akan di-*Link* ke program Java yaitu diasosiasikan dengan program saat jalan [11].

## **2.5 Faktor Sosial Ekonomi Keluarga Mempengaruhi Status Gizi Pada Balita**

Status gizi pada balita sangat rentan di pengaruhi oleh faktor eksternal diantaranya faktor sosial ekonomi keluarga antara lain pendapatan keluarga, tingkat pendidikan ibu, pengetahuan ibu mengenai pemantauan pertumbuhan anak, dan pengasuh anak [12]. Status gizi pada balita selain berpengaruh pada kesehatan balita juga sangat berpengaruh pada kecerdasannya. Balita yang memiliki status gizi rendah biasanya akan di iringi juga dengan rendahnya kecerdasan yang di miliki.

### **2.5.1 Penelitian Relevan**

Penelitian sebelumnya pernah di lakukan oleh Hadivian dan Sylvia dimana pendapatan keluarga di nilai sangat penting dalam pemenuhan asupan gizi pada balita, pendapatan keluarga biasanya akan berbanding lurus dengan terpenuhinya kebutuhan pangan serta gizi dalam keluarga.

Faktor tinggi atau rendahnya pendidikan ibu serta tingkat pengetahuan ibu juga sangat berpengaruh pada status gizi balita. Ibu dengan tingkat pendidikan menengah ke atas cenderung memiliki pengetahuan yang baik tentang asupan gizi. Peran orang tua sangat berpengaruh terutama pada ibu, karena seorang ibu berperan dalam pengelolaan rumah tangga dan berperan dalam menentukan jenis makanan yang akan dikonsumsi keluarganya. Ada pengaruh pengetahuan, sikap praktek ibu tentang gizi dan pendapatan keluarga terhadap tingkat kecukupan energi dan protein anak balita [13].

Penelitian juga di lakukan oleh Lutviana dan Budiono ada hubungan antara tingkat pendapatan dengan status gizi balita [14]. Berdasarkan latar belakang diatas maka peneliti ingin mengetahui apakah ada hubungan tingkat pendidikan ibu dan pendapatan keluarga terhadap status gizi anak.